บิ๊กจิ๋วแต่แจ๋ว

โปรแกรมลดขนาดแฟ้มขวัญใจนักพิมพ์ดีดได้ 5 เท่า

ประภาส จงสถิตย์วัฒนา

โปรแกรมขวัญใจนักพิมพ์ดีด เป็นเวิร์ดโปรสเซสเซอร์ที่มีอายุมากโปรแกรมหนึ่ง คือ เริ่มปล่อยสู่ท้องตลาดเมื่อปี พ.ศ. 2525 และมีการปรับปรุงแก้ไขมาโดยตลอด ปัจจุบันนี้ รุ่น 4.2 ก็นับได้ว่ามีความเชื่อถือได้ดีพอสมควร

การทำงานของโปรแกรม จะเก็บข้อมูลเอกสารเป็นแฟ้มเสมือน ซึ่งไม่ถูกจำกัดด้วยขนาดความจำ (ในเครื่อง IBM data segment จะจำกัดที่ 64 K ไบต์) ขนาดเอกสารจะใหญ่เท่าไรก็ได้ ซึ่งจะถูกจำกัดโดยขนาดจานแม่เหล็ก ถ้าเป็นฟลอปปี้ก็ 360 K ถ้าเป็นฮาร์ดดิสค์ก็ใหญ่ได้มาก การที่ทำเป็นแฟ้มเสมือนทำให้มีการอ่านเขียนข้อมูลจากจานแม่เหล็กบ่อย ผลที่ตามมาก็คือ เวลาเลื่อนหน้าขึ้นลงมากๆแล้วจะช้า ต้องรออ่านข้อมูลจากจานแม่เหล็ก แต่ถ้าเป็นฮาร์ดดิสค์ก็เร็วใช้งานได้ดี

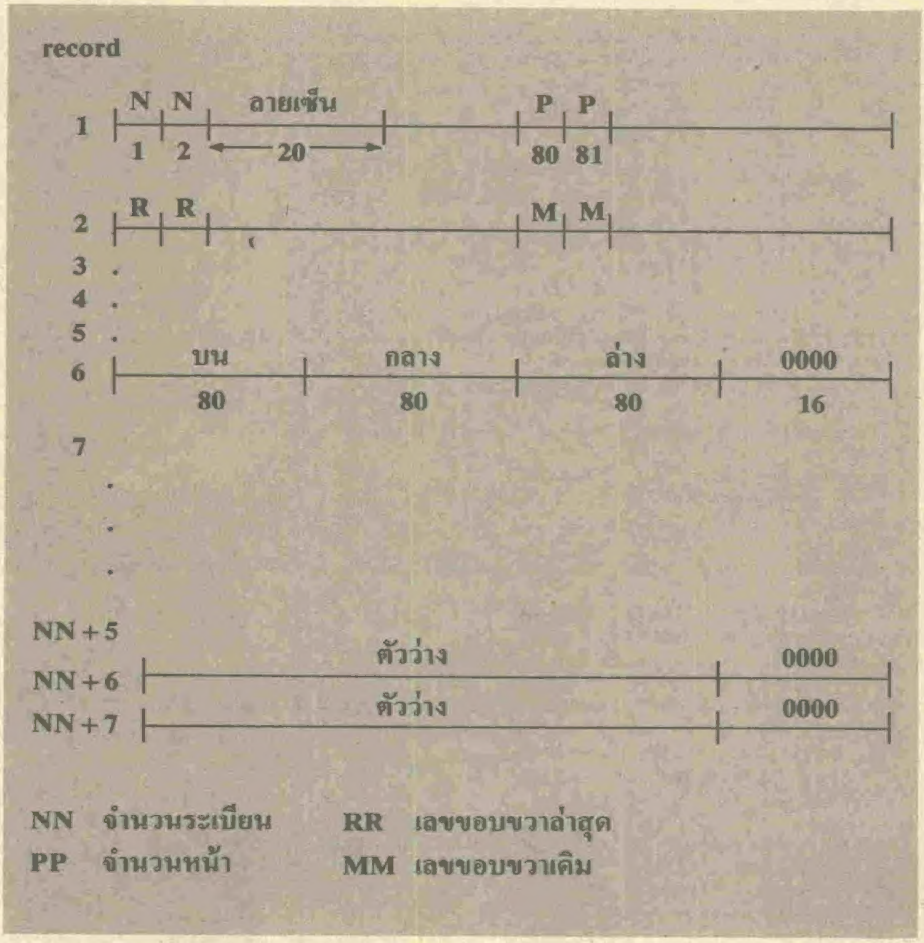
ผู้ใช้ขวัญใจนักพิมพ์ดีด จะพบว่าขนาดแฟ้มเอกสารที่เก็บบนจานแม่เหล็กมีขนาดใหญ่ โดยเฉพาะถ้าใช้ฟลอปปี้ดิสค์จะต้องคอยระวังดูว่า มีพื้นที่บนจานพอที่จะทำงานได้หรือไม่ ถ้าไม่พอ เมื่อใช้งานไปกลางคันอาจจะเกิด error disk full ทำให้อารมณ์เสีย อยากทุบเครื่องทิ้งก็ได้ *(ผมเจอมาหลายครั้งแล้ว เลยต้องมาเขียนโปรแกรมลดขนาดในบทความนี้)*

บิ๊กจิ๋วใช้ทำอะไร

ที่แฟ้มเอกสารมีขนาดใหญ่ก็เพราะโครงสร้างข้อมูลของขวัญใจนักพิมพ์ดีดเก็บสามบรรทัด *(จะอธิบายในตอนต่อไป)*

ตารางที่ 1 ขนาดที่ลดลง

แฟ้ม	ขนาด	เป็น TXT (ไบต์)	ลด (จำนวนเท่า)
expert.thi	15872	2393	6.63
microl.thi	30208	4320	6.99
software.thi	26368	4676	5.63
kumicro.thi	144384	28305	5.1



รูปที่ 1 โครงสร้างแฟ้ม .THI

เรามีวิธีที่จะลดขนาดได้โดยเปลี่ยนข้อมูลเป็นบรรทัดเดียว ผลจากการใช้งานโปรแกรมลดขนาดนี้ทำให้ข้อมูลลดลงได้ 5-7 เท่า *(ดูตาราง 1)* แฟ้มที่ลดขนาดแล้ว (ให้นามสกุล .TXT) เวลาจะเอามาแก้ไขต้องแปลงกลับไปเป็น .THI อย่างเดิมก่อน วิธีนี้จะเหมาะสำหรับการเก็บเอกสารเพื่อนำมาใช้ภายหลัง เอกสารที่ใช้ประจำมักจะไม่แปลง เพราะเสียเวลา

ทำได้อย่างไร

เราเปลี่ยนจากสามบรรทัดเป็นหนึ่งบรรทัดอย่างไร ก่อนอื่นต้องรู้จักโครงสร้างข้อมูลของแฟ้ม .THI เสียก่อน ดูรูปที่ 1 แฟ้ม .THI เป็น fixed length record สามารถทำ random access ได้ แต่ละระเบียน (record) มีขนาด 256 ไบต์ 5 ระเบียนแรกเป็นส่วนหัว เก็บข้อมูลเกี่ยวกับรูปร่างของเอกสาร ระเบียนที่ 6 เป็นต้นไปเป็นข้อมูลตัวหนังสือ

ข้อมูลที่ส่วนหัวก็มี

ระเบียนที่ 1

ไบต์ 1, 2 เก็บจำนวนระเบียนทั้งหมดของเอกสารนี้ (รวมส่วนหัวด้วย)

ไบต์ 80, 81 บอกจำนวนหน้าซึ่งคั่นโดย @PA (เอาไว้เลือกหน้าพิมพ์)

ระเบียนที่ 2

ไบต์ 1, 2 เก็บเลขตั้งขอบขวา (right margin) อันล่าสุด

ไบต์ 80, 81 เก็บเลขตั้งขอบขวาอันเดิมที่เริ่มสร้างแฟ้มนี้

ระเบียนที่ 3, 4, 5 ไม่ได้ใช้อะไร

ข้อมูลตัวหนังสือ เก็บขนาดตายตัว 3 บรรทัด ดูรูปที่ 2 บรรทัดละ 80 ตัวอักษร แบ่งเป็น 4 เซต คือ บน 80 ไบต์, กลาง 80 ไบต์, ล่าง 80 ไบต์, ว่าง 16 ไบต์ ทุกบรรทัดปิดท้ายด้วยรหัส 0D0A รหัสที่เก็บบรรทัดบนเป็นรหัสผสม (เช่น ะ ั ...) แบบรหัสเกษตร (ไม่น่าสงสัย

รูปที่ 2 ข้อมูลแบบสามบรรทัด

เลย) พอหมดตัวหนังสือแล้ว ก็มีระเบียนบรรทัดว่างๆอีก 2 ระเบียน

สังเกตไหมว่า ส่วนหัวเปลืองมากและค่อนข้างประหลาดอยู่ โดยเฉพาะระเบียนที่ 3 ตรงไบต์ 80, 81 มี A020 อยู่ ก็ไม่รู้ใช้ทำอะไร ส่วนระเบียนที่ 4, 5 มีเสียเปล่าแต่ไม่ได้ใช้งานอะไร จุดมุ่งหมายเดิมของผู้ออกแบบขวัญใจนักพิมพ์ดีดคงจะอยากเอามาใช้อะไรที่เราไม่รู้ มาถึงปัจจุบันนี้ก็ลืมกันไปหมดแล้ว

ร ะ ก ก ะ ง

รูปที่ 3 ข้อมูลแบบหนึ่งบรรทัด

(ความลับโครงสร้างข้อมูลนี้ ผมได้มาจากการ DEBUG อย่างอดทนและสนุกสนาน)

ตอนนี้หันมาดูแฟ้มแบบหนึ่งบรรทัดบ้าง (.TXT) ใช้แฟ้มแบบ text เก็บข้อมูล โดยไม่เอาตัวว่างบรรทัดบน-ล่างมา และไม่เอาตัวว่างด้านขวาของบรรทัดกลาง ดูรูปที่ 3 ลงท้ายแต่ละบรรทัดด้วย 0D0A

ส่วนที่ลดไปคือ ตัวว่าง บรรทัดบน-ล่าง และตัวว่างด้านขวา นั่นเอง

ส่วนหัวของ .TXT ต้องเก็บข้อมูลจากส่วนหัวของ .THI ไว้เป็นตัวหนังสือหนึ่งบรรทัด ดังนี้

\□ ลายเซ็น □ XXXX □ YY □ ZZ □ PPPP

XXXX เลขจำนวนระเบียน
YY เลขขอบขวาล่าสุด
ZZ เลขขอบขวาเก่า
PPPP เลขจำนวนหน้า
□ ตัวว่าง

ลายเซ็น 20 ตัวอักษร (จะกล่าวต่อไป)

เพื่อที่จะให้รู้ชนิดของแฟ้มอัตโนมัติ ผมเพิ่มลายเซ็น (signature) เข้าไปในส่วนหัวของแฟ้มทั้งสองชนิด คือ

แฟ้ม THI ลายเซ็น THI THAIEASY 4.2 KU

แฟ้ม TXT ลายเซ็น TXT THAIEASY 4.2 KU

วางไว้ที่ไบต์ที่ 3 เป็นต้นไป ดังนั้นการดูชนิดก็มองหาจากที่นี่ได้

โปรแกรม

ส่วนการแปลงอยู่ที่โพรซีเยอร์ TOTXT, TOTHI ลักษณะการทำงาน

รูปที่ 4 Listing ของโปรแกรม CV.PAS

```
program cv;
{ converter between .thi and .txt

  for education and recreation
  p. chongstitvatana
  last update          5 september 1987 }

const maxlen = 255;

type  thirec = array[0..maxlen] of char;
      thi    = file of thirec;
      txt    = text;
      aline = string[255];
      word  = string[20];

const thisig : word = 'THI THAIEASY 4.2 KU';  {signature}
      txtsig : word = 'TXT THAIEASY 4.2 KU';

var  buff,blank : thirec;
     buff2 : aline;
     fin  : thi;  fin2  :txt;
     fout : txt;  fout2 :thi;
     i,numrec,numline,numpage,rmargin,omargin :Integer;
     defaultflag :boolean;
     file1,file2,form :word;

procedure openread_thi(var inputfile : thi; filename :word);
begin
  assign(inputfile,filename);
  reset(inputfile);
end;

procedure openwrite_txt(var outputfile :txt; filename :word);
begin
  assign(outputfile,filename);
  rewrite(outputfile);
end;

procedure openread_txt(var inputfile : txt; filename :word);
begin
  assign(inputfile,filename);
  reset(inputfile);
end;

procedure openwrite_thi(var outputfile :thi; filename :word);
begin
  assign(outputfile,filename);
  rewrite(outputfile);
end;

procedure error( n :integer);
begin
  case n of
  1 : writeln('error character sequence of .txt file');
  end;
end;

procedure getword( var buffer :aline;
                   var w :word; var p :integer);
{ scan word from buffer, begin at position p
  return word w                                  }
var c :char; m: integer;
begin
  repeat
    c := buffer[p];
    p := p+1;
  until c <> ' ';        {skip leading blank}
  w := '';
  p := p-1;
  m := length(buffer);
  c := buffer[p];
  while (c<>' ') and (p<=m) do begin  {collect char}
```

รูปที่ 4 (ต่อ)

```
    w := w+c;
    p := p+1;
    c := buffer[p];
  end;
end;

procedure header_thi;
var i:integer;
begin
  read(fin,buff);        {read header}
  numrec := ord(buff[0]) + 256*ord(buff[1]);
  numline := numrec -5;
  numpage := ord(buff[80]) + 256*ord(buff[81]);
  read(fin,buff);
  rmargin := ord(buff[0]);
  omargin := ord(buff[80]);
  for i := 1 to 3 do read(fin,buff);
end;

function find_end(var buffer :thirec) :integer;
{ scan to find end of line }
var k :integer; c,cu,cd :char; stop :boolean;
begin
  k := 159;
  stop := false;
  repeat
    repeat       {find 0d0a}
      c := buffer[k];
      k := k-1;
    until (c=#10);
    if buffer[k] = #13 then stop := true;
  until stop;
  k := k-1;
  repeat            {skip blank at end of line}
    c := buffer[k];
    cu := buffer[k-80];
    cd := buffer[k+80];
    k := k-1;
  until (c<>' ') or (cu<>' ') or (cd<>' ') or (k<80);
  find_end := k-79;
end;

procedure totxt( var bthi :thirec; var btxt :aline);
var i,m :integer; c :char;
begin
  m := find_end(bthi);
  btxt := '';                  {clear btxt}
  for i := 0 to m do begin
    btxt := btxt + bthi[i+80];     {normal line}
    c := bthi[i];                {up line}
    if c <> ' ' then btxt := btxt + c;
    c := bthi[i+160];
    if c <> ' ' then btxt := btxt + c;
  end;
end;

procedure tothi( var btxt :aline; var bthi :thirec);
var i,k,endline :integer; c :char;
begin
  bthi := blank;
  k := 0;
  for i := 1 to length(btxt) do begin
    c := btxt[i];
    case ord(c) of
      32..127, 160..214 : begin
          bthi[k+80] := c ;       {normal line}
          k := k+1;
        end;
      215,216 : begin
          if k > 80 then error(1)
                    else bthi[k+159] := c; {dw line}
        end;
      217..254 :begin
          if k < 1 then error(1)
                   else bthi[k-1] :=c;     {up line}
        end;
    end; {case}
  end;   {for i}
  if (k-1) > rmargin then endline := k-1
                     else endline := rmargin;
  bthi[endline] := #13;
  bthi[endline+1] := #10;
  bthi[endline+80] := #13;
  bthi[endline+81] := #10;
  bthi[endline+160] := #13;
  bthi[endline+161] := #10;
end;

procedure makeblank( var buffer :thirec);
var i:Integer;   {rmargin is known}
begin
  for i := 0 to 239 do buffer[i] := #32;
  for i := 240 to 256 do buffer[i] := #0;
end;

procedure default;
{ if no header in txt file then
  make default value of numrec and rmargin }
begin
  reset(fin2);          {back to begin of file}
  defaultflag := true;
  numrec := 5;
  rmargin := 75;
  omargin := 75;
end;

procedure header_txt;
var buff :aline; p,dummy :integer; w :word;
begin
  readln(fin2,buff);
  if length(buff) <= 0   {blank line}
  then default
  else begin
       if buff[1] <> '\'
       then default
       else begin
              p := 23;    {start position of numrec}
              getword(buff,w,p);       {get numrec}
              val(w,numrec,dummy);
              getword(buff,w,p);       {get rmargin}
              val(w,rmargin,dummy);
              getword(buff,w,p);       {get omargin}
              val(w,omargin,dummy);
              getword(buff,w,p);       {get numpage}
              val(w,numpage,dummy);
            end;
     end;
end;

procedure writeheader_thi;
var head : thirec; i :integer;
begin
  head := blank;
  head[0] := chr(lo(numrec));
  head[1] := chr(hi(numrec));
  for i := 1 to length(thisig) do
    head[i+1] := thisig[i];       {signature}
  head[80] := chr(lo(numpage));
  head[81] := chr(hi(numpage));  {number of page}
  write(fout2,head);
  head := blank;
  head[0] := chr(lo(rmargin));  {right margin}
  head[1] := chr(hi(rmargin));
  head[80] := chr(lo(omargin));  {old right margin}
  head[81] := chr(hi(omargin));
  write(fout2,head);
```

รูปที่ 4 (ต่อ)

```
  head := blank;
  head[80] := #$a0;
  head[81] := #32;              {I don't know ??}
  write(fout2,head);
  write(fout2,blank);
  write(fout2,blank);           {2 blank records}
end;

procedure writeheader_txt;
begin
  writeln(fout,'\ ',txtsig:20,' ',numrec:4,' ',
          rmargin:2,' ',omargin:2,' ',numpage:4);
end;

procedure thitxt;  {thi-->txt}
var i :Integer;
begin
  openread_thi(fin,file1);
  header_thi;
  writeln('number of line ',numline);
  makeblank(blank);
  openwrite_txt(fout,file2);
  writeheader_txt;
  for i := 1 to numline do
  begin
    read(fin,buff);
    totxt(buff,buff2);
    writeln(fout,buff2);
  end;
  close(fin);
  close(fout);
end;

procedure rewriteheader;
var head : thirec; i :integer;
begin
  seek(fout2,0);    {to first record}
  head := blank;
  head[0] := chr(lo(numrec));
  head[1] := chr(hi(numrec));
  for i := 1 to length(thisig) do
    head[i+1] := thisig[i];       {signature}
  head[80] := chr(lo(numpage));
  head[81] := chr(hi(numpage));
  write(fout2,head);
end;

procedure txtthi;
begin                    {txt --> thi}
  openread_txt(fin2,file1);
  header_txt;
  if not(defaultflag) then begin
    numline := numrec -5;
    writeln('number of line ',numline);
  end;
  makeblank(blank);
  openwrite_thi(fout2,file2);
  writeheader_thi;
  numrec := 5;              {first five are header}
  while not(eof(fin2)) do
  begin
    readln(fin2,buff2);
    tothi(buff2,buff);
    write(fout2,buff);
    numrec := numrec +1;
  end;
  close(fin2);
  write(fout2,blank);
  write(fout2,blank);
  if defaultflag then rewriteheader;
  close(fout2);
end;
```



```
procedure init;
begin
  defaultflag := false;
end;

procedure getform(inputfile :word;
                  var form :word);
{ open inputfile and read 128 char to see
  what type it is.  return THI TXT in form }
var f : file;  {untype}
    rbuff : array[1..128] of byte;  {read buffer}
    w,x :word; i :Integer;
    ok :boolean; c :char;
begin
  assign(f,inputfile);
  {$i-} reset(f) {$i+} ;      {check for existance}
  ok := (ioresult = 0);
  if not(ok)
  then begin
    writeln('cannot find file ',inputfile);
    halt; end
  else begin
    blockread(f,rbuff,1);
    close(f);
  end;
  w := '';
  for i := 1 to 3 do w[i] := upcase(rbuff[i+2]);
  w[0] := chr(3);
  x := 'NIL';
  if w = 'THI' then x := w;
  if w = 'TXT' then x := w;
  if w = '   ' then x := 'THI';
  if x = 'NIL' then x := 'TXT';
  form := x;
end;

procedure cut_ext(inname :word;
                  var outname :word);
var i :integer;
begin
  i := pos('.',inname);
  outname := copy(inname,1,i-1);
end;

procedure makename( firstname,form :word;
                    var secondname :word);
var f1,ext :word;
begin
  cut_ext(firstname,f1);
  if form = 'THI' then ext := '.TXT';
  if form = 'TXT' then ext := '.THI';
  secondname := f1 + ext;
end;

procedure confirm(file1,form,file2 :word);
var f :file; ok :boolean; c :char;
begin
  if form = 'THI' then begin
    writeln(file1,' --> ',file2);
    writeln('[ THI -> TXT ]');
  end;
  if form = 'TXT' then begin
    writeln(file1,' --> ',file2);
    writeln('[ TXT -> THI ]');
  end;
  assign(f,file2);   {writecheck file2}
  {$i-} reset(f) {$i+};
  ok := (ioresult = 0);
  if ok
  then begin          {old file exists}
    close(f);
    write('overwrite old ',file2,' y/n ? ');
```

รูปที่ 4 (ต่อ)

```
    read(kbd,c); writeln(c);
    if upcase(c)<> 'Y'then halt; end
  else begin
    writeln('esc to break, any key to continue');
    read(kbd,c);
    if c=#27 then halt;
  end;
end;

var n :integer;
{main}
begin
  writeln;
  writeln('Converter v1.0  KU micro lab  1987');
  writeln;
  n := paramcount;
  case n of
  0 : begin
        write('input file name : ');
        readln(file1);
        write('output file name: ');
        readln(file2);
        getform(file1,form);
        if file2='' then makename(file1,form,file2);
      end;
  1 : begin
        file1 := paramstr(1);
        getform(file1,form);
        makename(file1,form,file2);
      end;
  2 : begin
        file1 := paramstr(1);
        getform(file1,form);
        file2 := paramstr(2);
      end;
  end;
  confirm(file1,form,file2);
  if form = 'THI' then thitxt;
  if form = 'TXT' then txtthi;
end.
```

ครั่วๆคือ อ่านข้อมูลมาทีละบรรทัด แล้วแปลง สุดท้ายก็จะเขียนลงแฟ้มออก TOTXT เริ่มจากการตัดตัวว่างขวามือ โดยมองหาขวาของหนังสือทั้งสามบรรทัดใน .THI ด้วย FIND-END ตอนแรกมองจากขวาสุด ตัวที่ 80 ก่อน ถอยมาหา 0A แล้ว 0D แล้ววิ่งเข้าไปจนเจอตัวหนังสือ TOTXT แปลงโดยดึงแถวกลางก่อน (ว่างไม่ว่างเอาหมด) ดึงบน-ล่างถ้าไม่ว่าง จนไปถึงขวาที่หาไว้

TOTHI เริ่มจากทำบรรทัดว่างมา 3 บรรทัดก่อน เป็นระเบียนขนาด 256 ไบต์ ตาม .THI ดึงจาก .TXT มาทีละตัว ใส่บรรทัดบน กลาง ล่าง ตามแต่กลุ่มรหัส และปิดท้ายทุกบรรทัดด้วย 0D0A

เรายอมให้ .TXT ที่จะแปลงนี้ไม่ต้องมีส่วนหัวก็ได้ เพื่อสามารถเอามาจากแฟ้มเอกสารของเวิร์ดโปรเซสเซอร์อื่นๆ เช่น ราชวิถี, Sidekick ได้ โดยตรวจสอบตัวแรกของแฟ้ม .TXT ว่าเป็น \ หรือไม่ ถ้าไม่มีส่วนหัว (บอกที่ default flag) จะใช้ค่าต่างๆตาม default คือ ขอบขวา 75, จำนวนหน้า 1 แต่จำนวนระเบียนจะรู้ได้เมื่อแปลงไปจนจบแฟ้ม .TXT แล้ว ดังนั้นต้องแปลงจนหมด พร้อมกับนับจำนวนบรรทัดไป แล้วย้อนกลับมาใส่ส่วนหัวของ .THI อีกที โดย REWRITE HEADER

โปรแกรมจะตรวจดูชนิดของแฟ้มว่าเป็น THI หรือ TXT โดยดู 3 ตัว เริ่มจาก ไบต์ที่ 3 ด้วยการเปิดแฟ้มแบบไม่มีชนิดอ่านมา 128 ไบต์ เอามาดู ถ้ามีลายเซ็นอยู่ ก็รู้ชนิดแน่แน่ ถ้าไม่ถือว่าตัวว่าง 3 ตัว เป็นชนิด .THI (ของเดิมเป็นเช่นนี้) นอกจากนั้น ถือเป็น TXT ถ้าจะมีชนิดอื่นๆก็ไปแก้ที่ GETFORM เองครับ

การใช้งาน

เรียก CV แล้ว จะใส่ชื่อแฟ้มไปเลยก็ได้ ถ้าไม่ใส่ชื่อแฟ้มโปรแกรมจะมาถามเอง ชื่อแฟ้มขาออกไม่ใส่ก็ได้ ซึ่งจะ default เป็นชื่อแฟ้มขาเข้า เปลี่ยนเป็นนามสกุล .TXT ให้ วิธีเรียกทำได้ดังนี้

CV [แฟ้มเดิม] [แฟ้มใหม่]

[ ] = optional

เนื่องจากโปรแกรมจะตรวจชนิดของแฟ้มเอง จะรู้ว่าต้องเปลี่ยน THI→TXT หรือ TXT→THI โดยเราไม่ต้องบอก สะดวกดี

ส่งท้าย

ความจริง (บิ๊ก) จิ๋วแต่แจ๋วนี้ มีอยู่ในขวัญใจนักพิมพ์ดีด รุ่น 4.2 อยู่แล้ว แต่ฟังก์ชันการแปลงอันนั้น ยังมีข้อผิดพลาดอยู่ ในบางครั้งแปลงไปกลับจะหายไป 2 บรรทัดสุดท้าย และใน .TXT ไม่เก็บข้อมูลพวกตั้งขอบขวาเลย ทำให้เวลาแปลงกลับแล้วไม่เหมือนเดิม

ผู้เขียนขอขอบคุณ *วิวรรธน์ อิ่มอารมณ์* ผู้มีความอดทน บำรุงรักษาขวัญใจนักพิมพ์ดีดด้วยดีตลอดมา